Run-time planning of case-based business processes

Danillo Sprovieri, Daniel Diaz, Raúl Mazo, Knut Hinkelmann

▶ **To cite this version:**

Run-Time Planning of Case-based Business Processes

Danillo Sprovieri, Daniel Diaz, Raúl Mazo
Centre de Recherche en Informatique
Université Paris-Sorbonne
Paris, France
{danillo.sprovieri, daniel.diaz, raul.mazo}@univ-paris1.fr

Knut Hinkelmann
Institute of Information Systems, School of Business
University of Applied Sciences and Arts Northwestern
Switzerland
Olten, Switzerland
Knut.hinkelmann@fhnw.ch

Abstract— Context: **Organizations act in highly competitive markets, which forces them to be flexible. Constantly changing business requirements require flexible business processes. Case Management Model and Notation (CMMN) supports modeling run-time flexibility of partially structured business process models, but does not fully specify the control flow.** *Objective:* **The goal is to develop a planning algorithm that supports the case worker in planning case-based business processes at run-time.** *Method:* **We identify the requirements of run-time planning of partly structured processes by analyzing the admission process for the master degree at FHNW. To plan the process instance, we develop a planning algorithm. Our planning algorithm is evaluated using concrete cases provided by FHNW in order to demonstrate real application.** *Results:* **The planning algorithm reflects the requirements for serializing tasks at run-time.** *Conclusion:* **Our planning algorithm allows to automatically deriving context-specific execution plans for CMMN models at run-time.**

Keywords—Artificial Intelligence; Business Process; CMMN; Planning

I. INTRODUCTION

A common consequence of constantly changing business requirements is the need of flexible business processes. Business processes need to be adapted continuously. However, the adaption of business processes is a challenging, time consuming and error-prone task. In the last decade, there has been an increased use of business process management (BPM) tools by enterprises and emerging standards for business process specifications and execution (e.g. BPMN [1] and WS-BPEL [2]). These techniques support businesses in modelling and executing business processes. Commonly, BPMN is used to model structured business processes. BPM along with modelling of business processes supports quick reaction to environmental changes. This can be achieved by adapting the process to new circumstances. But BPMN models have their limitations regarding flexibly adaptation of their processes. While BPMN models allow re-engineering the process at design-time, they do not support flexible business process execution (at run-time).

The definition of flexibility is not standardized and is interpreted differently in the literature. We refer to the definition of Dadam et al. [3]. Flexibility at design-time enables quick creation of new business process models at design-time using notation languages like BPMN. In contrast,

flexibility at run-time enables agile deviation of the planned sequence or to decide for an execution plan only at run-time. More precisely, the process instance can be individually modified (e.g. switching, shifting, or deletion of process steps). Since process models are subjects of constantly changing nature, process changes are required. In addition, we use the classification of processes according to Gadatsch [4] (structured processes, case-based processes, and ad hoc processes). Structured process models cannot possibly consider all feasible variations of the case-based and ad hoc process at design-time; more particularly, it is not possible to anticipate all possible situations. Structured processes, case-based processes and ad-hoc processes can contain knowledge work [5]–[7]. Knowledge-intensive processes are neither fully predictable nor repetitive, and therefore cannot be fully pre-specified at design-time. The unpredictability of these processes demands a certain amount of looseness [8]. Reichert & Weber [8] characterize loose business processes as unpredictable, non-repeatable and emergent. The exact course of action only emerges during process execution (i.e. at run-time) when more information becomes available. Only the goal of these processes is known a priori. The parameters (e.g. rules, conditions and events) determining the course of action are typically known at design-time and might change during process execution. Correspondingly, knowledge-intensive processes cannot be fully pre-specified and require a loose specification.

Due to the missing looseness property, models in BPMN are not flexible, since the BPMN specification does not allow arbitrary sequence of optional activities. Looseness is one of the main requirements of flexible business processes along with variability, adaption, and evolution [9]. Process flexibility is required and can be seen as the ability to deal with both foreseen and unforeseen changes [10]. The need for process flexibility has been also recognized in the workflow and process technology communities in order to support organizations to adapt to changing business requirements [11], [12]. Correspondingly, several research trends try to express, analyse and support the dynamic adaption of business processes at run-time [13]. These trends foster dynamic adaptation of business processes by deleting, modifying, or adding one or several process elements, or by changing the order of elements at run-time. One common property of these approaches is that they do not focus on the control flow of the

process. The control flow is determined at run-time and depends on the context during process execution.

An approach to deal with the flexibility issue is Adaptive Case Management (ACM) [14]. ACM provides organizations benefits such as improvements to knowledge worker effectiveness, improved responsiveness, and flexibility in adapting to market changes [15]. ACM does not focus on the flow of control, as it presumes it cannot be defined at design-time. The control flow is partly defined by the context at run-time. The current status of information, events, conditions, rules and human judgment determine how the case is executed. CMMN is a modelling language developed by OMG [16], which is based on principles of ACM and supports run-time adaption by loosely coupled tasks and presenting them to the caseworker based on pre-defined conditions. Thus, CMMN fosters flexibility of processes by giving guidance on which task can be executed under given conditions (e.g. in form of sentries or planning tables). Planning in CMMN is done by presenting applicable tasks using the context: a worklist is presented to the caseworker containing all tasks that can be executed in the current state of the process. The context is evaluated and defined if a task is put on the worklist. CMMN distinguishes two types of tasks: ordinary and discretionary tasks. The latter enables flexibility during process execution. It is up to the caseworker to select which of the discretionary tasks should be executed and therefore the caseworker brings the tasks into sequence. CMMN captures preconditions of discretionary tasks using (applicability) rules, which define the applicability of a task depending on the context. CMMN models try to catch the knowledge about process execution known at design-time. CMMN does not fully specify the process. While it specifies the existence of a worklist, it does not provide an algorithm on how this worklist is initially populated and maintained during the execution of the business process. It is possibly to specify the sequence, but it cannot be visualized in CMMN models. There is a research gap to better support the run-time planning of the case worker. The idea is to support run-time activity selection for flexible case-based business processes, specifically addressing the new OMG standard, CMMN.

In Section 2, we present a concrete business process to show practical usage. In Section 3, we elaborate our solution proposal and describe the planning algorithm to be used on CMMN models. In Section 4, we present a preliminary evaluation and in Section 5 position our contribution in relation to previous approaches. We conclude in Section 6, discuss the results, and give indication for further research.

II. CASE

In this Section, we analyse and identify the requirements of the problem to plan a non-structured business process at run-time in a concrete situation. This is based on a literature review and an examination of a real business process with non-structured parts. The business process describes the admission process of a candidate for the study of a master degree provided by the School of Business FHNW. The aim of the process is to select suitable candidates for the master degree program. In the case of the admission process, the fundamental questions are how and why the applicant can be accepted or rejected; also, how the process execution can be improved. The answers to these questions provide background knowledge of the motivating scenario.

The focus is on the sub-process[1] (Figure 1) prepare eligibility check, since it is a knowledge intensive task and the control flow not fully structured. The sub-process prepare eligibility check is executed by the case worker. After an application for a Master of Science program arrives, the case worker sends a confirmation to the applicant and prepares the eligibility check. The case worker checks, whether the candidate is eligible, and starts to check the completeness of documents. Depending which documents of the candidate are available during the execution of check completeness of documents (i.e. at run-time), the case worker must execute the following tasks: check the bachelor degree, check the accreditation of the university, calculate the average grade, and check if grade is at least good. The tables symbolized on top of some tasks represent planning tables. Applicability rules are listed in the planning table. When the rule for a discretionary task (e.g. check Transcript of Record) is true, this task waits for execution on the worklist, but it is up to the case worker to decide at run-time, if its execution is required. Typically, applicability rules indicate the case worker discretionary tasks, which "can" be, but do not need be executed considering a specific context. In the sub-process prepare eligibility check, the case worker can anytime over the whole sub-process execution discuss the bachelor degree with the dean, if the case worker needs and wants further clarification to check the bachelor degree. The execution of these tasks varies depending on the experiences and competences of the case worker. The green arrow represents a link to a model that describes the planning table. In the case plan, all (ordinary) tasks have pre-conditions symbolized as a sentry. The sentry is represented as a blank diamond. For instance, the sentry for the task check completeness of documents is activated by the event "application is arrived". A sentry is activated once a condition and/or an event occur. If so, the task is put on the worklist and waits for execution. In contrast to discretionary tasks, ordinary tasks once enabled and waiting on the worklist for their
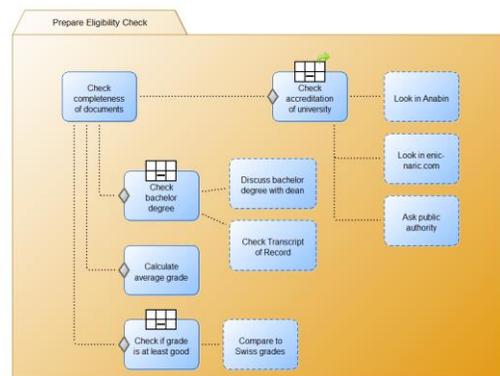

Figure 1. Case Plan Model of the sub-Process Prepare Eligibility Check

---

[1] The Case Plan is designed with the Knowledge Work Designer developed at FHNW by Hinkelmann and Pierfranceschi [29], [30]. The knowledge designer offers the integration of BPMN [1], CMMN [16] and DMN [31].

execution, must be executed. The activation of the sentry depends on its entry criteria. A sentry criterion is defined by events that occur during run-time (on: <event>) and by conditions (if: <condition>). Applicability rules specify under which circumstances a discretionary task is enabled and put on the worklist.

A. Case Analysis

In this Section, we analyse the issues found in the case to derive at run-time the worklist of the sub-process prepare eligibility check and compare them to the problems found in the literature. The aim is to show that the case plan reflects the issues that have been found during literature review. We show that the case presented in the previous section is a good instrument to identify requirements for the developed planning algorithm. We conclude if the case plan can be used to build our planning algorithm on top of it. As we will outline, the described case is representative.

One of the main characteristics of the case is that the process flow can only be determined at run-time. Because the information needed (i.e. context) to determine the control flow is only given at run-time and depends on the individual case worker (e.g. competences, experience, discretion), it is not possible to define the sequence of tasks at design-time. It is not possible to represent discretionary tasks in a BPMN model. But if we would represent only ordinary tasks, the case plan can be represented in BPMN. In this case, the process model is fully structured and supports no flexible deviation of the workflow during run-time. Executable tasks and the flow of control are defined at design-time, so at run-time the case worker has no freedom to plan the process instance, since BPMN offers no possibility to represent discretionary tasks. Gateways and events are used in BPMN to define the control flow. These elements correspond to sentries in CMMN. They can be seen as Event-Condition-Action (ECA) rules.

The process flow also depends on the context in terms of competences, experiences and discretion of the case worker. It is up to the case worker which tasks are for instance required to be performed after the task check accreditation of university; whether additional discretionary tasks (e.g. to ask public authority, look in Anabin, and enic-naric.com) are necessary. If the case worker already knows the university where the candidate comes from, he does not need to access further information resources. But if the case worker doesn't know the university and the available information resources provided by FHNW aren't sufficient, the case worker can decide to execute additional discretionary tasks in order to get the required information (e.g. look in Anabin). Since the sequence of tasks is not pre-defined, it is determined at run-time which tasks can be executed at any given state. Certain tasks can only be executed after a certain change in the context of other tasks, and thus can depend on each other. If, for example, the case worker finds out during the execution of the task look in Anabin that the university, where the applicant got it from, is accredited, the tasks look in enic-naric.com and ask public authority are not required to be executed anymore. The changing context during run-time and the manual planning of the case worker are typical characteristics for the problem domain. CMMN is a suitable language for knowledge intensive processes and partly structured business processes. CMMN allows to describe the case in a loosely structured manner and supports partly the case worker in planning the process instance (by managing the task appearing on the worklist). Also, CMMN supports run-time adaption by loosely coupling tasks and presenting them to the case worker.

The case plan analysis indicates different situations how the case can be planned and how the worklist is updated. In the presented case plan, there are two different types of tasks that can be put on the worklist and differ about the judgment of their execution. On the one hand, (ordinary) tasks must be executed once enabled (e.g. if the condition of the sentry is evaluated true). On the other hand, the execution of discretionary tasks depends on the case worker's competences, experiences and/or proclivity. For the case worker to be allowed to decide if he wants to execute the task, applicability rules must be evaluated true. So, the derivation of these tasks on the worklist requires the true evaluation of the related conditions. As soon as a task is enabled, the task instance waits for a caseworker to be executed. If once an ordinary task is enabled, the task instance is waiting on the worklist for a case worker to be mandatorily executed. In contrast, if once a discretionary task is enabled, the case worker decides its execution according to the given context. The execution is not mandatory. Discretionary and (ordinary) tasks are types of PlanItems and can be in different states during process execution. These states are called lifecycles in CMMN and impact the update of the worklist. In total there are nine lifecycles describing the behaviour of a given state (e.g. active, enabled, terminated) (OMG, 2014). In CMMN, events occur during process execution, which facilitate the update of the worklist. Events may trigger for instance, the enabling or activation of a task, the achievement of a milestone, or the termination of a stage. Any event must have a cause and might change the worklist. CMMN [16] specifies events and their causes into two categories: events on PlanItems (e.g. task, eventlistener, milestone and planfragment) and on CaseFiles. Depending on the state of a given PlanItem, the worklist is updated. For instance, the termination and enabling of tasks can trigger further updates of the worklist. These states cause events that can for instance activate a sentry of the related task, so the pre-condition of the execution of this task is evaluated. As soon the task check completeness of documents is terminated, the sentries connected to this task are evaluated. If the event part of the sentry is evaluated true (on: check completeness of documents) and there is no condition or the condition is also evaluated true, the following task is selected for the worklist.

III. SOLUTION DESIGN

Based on the insights of the previous phase, we suggest and develop a planning algorithm that supports the case worker in updating and planning the worklist. We analyse and validate the planning algorithm against the problem description, while comparing similar solutions found in the literature. This guarantees that the aim of the proposed algorithm is to assist the case worker in planning the tasks in a flexible business process.

The evolution of the status of information allows CMMN models to represent a very broad range of business processes. In some business processes, it is not necessary to use such a broad frame. For instance, there are certain business processes that are executed in an environment, where some point in time exists at which the result of a task does not impact the availability of others. In these environments, a sequence of tasks to be executed can be derived in advance, since the change of context due to execution of further task will not affect the base for the decision anymore. There is a specific point of time in which the status of the information required for the process execution is available, so that the outcome of each task does not affect the worklist anymore. For this, all information required for the task execution must be known at a specific point in time during process execution. This guarantees an optimal execution trace based on pre-defined effects, impacts, conditions, rules and events at run-time [17].

According to the case worker's competences, experiences and discretion the process execution can vary. The goal is context-dependent and known a priori (i.e. at design-time). The goal of the admission process is to make a decision of admission. Using an appropriate search algorithm, the solution in form of a plan with a sequence of action is obtainable. In our approach, we focus on situations, where possible activities are known in advance (i.e. at design-time); but the correct use of these (i.e. selection and sequencing) is to be defined at run-time (i.e. depending on their context).

CMMN models must be instantiated in order to be able to plan the execution trace, since the information needed for the update of the worklist is obtained only during run-time. Depending on the context, the worklist is derived and the case worker partly decides at run-time which tasks from the worklist are executed. The worklist can only be updated, if the current status of information allows the selection of executable tasks. The selection is based on the following CMMN components: events that occur during process execution, sentries and planning tables, which must be evaluated. The worklist is not ordered and can be further planned by the case worker. In fact, CMMN defines the planning as the case worker's decisions to execute given enabled discretionary tasks. It would make sense to plan ordinary tasks order of execution, if dependencies exist. If the entry criteria of one ordinary task depends on the execution of another ordinary task, the tasks must be included in the planning. For this reason, the sequence in which these tasks can be executed is determined at design-time. But, it is only possible during run-time to evaluate if such dependency exists, as the dependency is context-specific. Conditions represented in sentries allow to represent this kind of dependency. If there is no dependency, the sequence of ordinary tasks can be arbitrary chosen. The sequence in which these ordinary tasks are executed does not impact the process execution in terms of cost or efficiency.

Using events on PlanItems and CaseFiles, a planning algorithm is able to select suitable tasks for the worklist at run-time. The current context needs to be evaluated in order to decide which of all available tasks can be selected. In order to develop our proposed solution, all CMMN components that impact the control flow must be considered to plan the case-based business process at run-time. We define an algorithm

that automatically updates the worklist. To do so, the algorithm considers the execution semantics of CMMN models and is able to evaluate the current status of information at run-time. CMMN models can be systematically analysed and used to support the caseworker in planning the execution of tasks at run-time. We develop a planning algorithm, which supports the case worker in selecting adequate tasks at run-time, while reusing existing and suitable CMMN models.

```
input: a CMMN model
result: adequate process execution

begin
        initialize W with the set of enabled
tasks
        while W is not empty
                let T be an ordinary task from W
                execute T
                update W // NB: also removes T
from W
                let L <- T₁,…,Tₙ the n discret.
tasks of T, which are also in W

                while L is not empty
                        ask the user to choose one
                task Tx in L to execute
                        execute Tx
                        update W // NB: also
                removes Tx from W

                        remove all tasks in L
                which are no longer in W // NB:
                also removes Tx from L

                end while
        end while
end
```

Listing 1. Planning Algorithm

We elaborate the proposed solution according to the results of the previous section. We develop a planning algorithm that supports case workers at run-time to update and plan the worklist. In Listing 1, a superordinate procedure is presented, which formalizes our planning algorithm. This demonstrates how the planning method is applied on a process instance. Furthermore, the algorithm indicates the importance of discretionary tasks in our planning method.

A change in the lifecycles of PlanItems and/or CaseFile triggers the re-evaluation of the conditions (of sentries and applicability rules). The basis for planning the sequence of both tasks calculate average grade and check if the grade is at least good is specified at run-time: only when the conditions are evaluated true, tasks can be put on the worklist. A sentry and planning table thus represent conditions for planning. The planning is based on events and conditions that must be evaluated. Depending on the current context, tasks are enabled and put on the worklist.

IV. PRELIMINARY EVALUATION

In order to show practical relevance of the planning algorithm, we use concrete process instances of the case-based business process prepare eligibility check and apply the planning algorithm literally. The results of the evaluation must

be considered with caution, since the planning method is only applied on case-based business processes within the domain of the public administration (PA). Further research is required in order to evaluate if the proposed method can be applied on other domains, and if it can be generalized further. We plan to do a large and complete validation of our approach, which is already included in our work agenda. The evaluation showed context-specific planning of CMMN models by automatically updating and planning the worklist at run-time. The time needed to plan discretionary PlanItems decreases. This means, the time of the case worker can be allocated for other business related tasks. The planning method supports the case worker partly, since the case worker needs to sustain its flexibility to execute discretionary tasks. Instead of proposing a list of executable PlanItems, our planning method proposes a more adequate sequence of tasks, which improves the execution of the admission process and decreases the cost for its execution.

V. Related Work

One approach to improve the ability of business processes to respond to changes is a Process-Aware Information System (PAIS). Flexibility requirements are integrated in the concept of a PAIS. It supports flexibility of business processes and is characterized by four flexibility requirements: variability, looseness, adaption, and evolution [9]. PAISs focus on the support of predictable and repetitive business processes, and provide more flexibility within specific processes. These systems have in common that there is an explicit process notion and that the information system is aware of the processes it supports. Even if processes are fully pre-specified, flexibility is required to support dynamic process adaptions in order to handle business process variability at run-time. Variability is defined as different process variants that result depending on the given context [18], [19]. PAISs are also used to support less structured processes, which are commonly called knowledge-intensive processes. To configure the business process at run-time, pre-defined skeleton processes are used. The PAWS [20] framework uses run-time mechanisms to flexibly compose business processes. The approach models business processes using BPEL, which requires a process skeleton to specify the control flow of the process at design-time. For this skeleton, candidates of executing services are selected during run-time. Our approach contrasts to the PAWS [20] framework, as we apply the context (e.g. context-specific goals) at run-time. Hence, we do not need a sophisticated analysis of possible executions at design-time. GEPSIS [21] system differs from our approach, as it does not support the user to plan the business process at run-time. Instead, it fosters the flexibility of the process model at design-time by offering a large variety of different business processes. GEPSIS supports to analyse the static structure of the process and allows quantitative evaluation of the dynamic behaviour. Promising approaches to manage flexibility are based on business artefacts. Hull et al. [22] proposed a formal semantics of the Guard-Stage-Milestone (GSM) focusing on the interaction between artefacts instance cases. Their contribution consists of showing the equivalence of three different formulations of the GSM semantics for artefact instance iteration. Formulations are based on ECA rules, mathematical properties, and first-order logic formulas. To support run-time adaptation of business processes, one possible approach is to foster variability in pre-specified business process models (i.e. configuration of process model at run-time). A practical example is presented in [23], which uses a process model repository with large collection of process variants (i.e. process families). Process variants can be represented through a configurable base process model (e.g. standardized processes, most frequently used process variants) and a related set of pre-specified changes (e.g. adjustment points). A similar approach to process families is used in software product line (SPL). In a SPL, information systems share a commonly managed set of features satisfying specific needs [24]. Another possibility to support run-time adaptation is Case-based Reasoning (CBR) [25]. CBR is a problem-solving method, which facilitates the reuse of experience of previous cases. Instead of composing complete new solution from scratch, CBR offers methods for representing, storing, indexing, retrieving, and adapting cases. Process participants benefit from solutions and experiences from similar situations in the past. Also, case handling, presented by Van der Aalst et al. [26], was introduced to support flexible business processes, but does not use predefined processes control structures to determine what should be done at run-time. It focuses on presenting tasks to the caseworker under certain circumstances. To foster run-time flexibility, Murguzur et al. [27] presented a context variability modelling approach. They also presented fundamental building blocks of a framework for enabling context variability in service-based DSPLs. Additionally, they presented in [28] a holistic methodology to automatically resolve process variability at run-time. The proposed solution performs a staged configuration considering dynamic context data to accomplish effective decision-making

VI. Conclusion

As we have outlined, design-time planning is not suitable for flexible case-based business processes, especially for the presented admission process. This is due to the fact that the information required for the execution of this process is provided at run-time. More precisely, the required information to decide, if the candidate is eligible is only available at run-time. CMMN allows to model such flexible process models by loosely coupling PlanItems. We presented a method for run-time planning of CMMN models to support the case worker's decisions.

A. Contribution

In this work, we contribute a new planning method for supporting case workers to achieve a context-specific goal. Especially for discretionary tasks, our planning method provides the support for the choice of tasks to be executed during run-time. We contribute a superordinate procedure, which binds together the updating of the worklist with the selection of ordinary and respective discretionary tasks, and describes how these PlanItems are passed into the algorithm in order to get an execution plan. To formalize the planning method, we developed an algorithm to plan a given CMMN model.

B. Future Work

As future work, we will continue our research by elaborating an extension of the proposed planning method to allow definition of new PlanItems at run-time. Current approaches do not incorporate options to formalize ad hoc processes, which can be stored and re-used for future business processes. The case worker has the possibility to create new PlanItems at run-time in order to support further process instantiations. Besides, we plan to better integrate the current context of the execution to allow a more dynamic planning. This can be useful for knowledge intensive processes (e.g. innovation processes). Furthermore, evaluation should be performed on a broader base in order to assess if the method can be applied to other domains. A larger sample size would also allow comparing the performance of a case worker supported by the method against a traditional setting. Furthermore, the method could be improved to also include the skillset of the case worker to propose more adequate discretionary tasks comparing different skillsets of other case workers. Similar skillsets would lead to similar discretionary tasks proposed.

REFERENCES

[1] OMG, "Business Process Model And Notation Specification 2.0," 2011. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/. [Accessed: 11-Apr-2016].

[2] OASIS, "Web Services Business Process Execution Language Version 2.0," 2007. [Accessed: 11-Apr-2016].

[3] P. Dadam, M. Reichert, and S. Rinderle-Ma, "Prozessmanagementsysteme," Informatik-Spektrum, vol. 34, pp. 364–376, 2011.

[4] A. Gadatsch, Grundkurs Geschäftsprozess-Management, 7th ed. Wiesbaden: Vieweg+Teubner Verlag, 2013.

[5] K. Hinkelmann, "Integration of BPMN and CMMN," OMG Technical Meeting, 2015. .

[6] T. Debevoise and J. Taylor, The MicroGuide to Process and Decision Modeling in BPMN/DMN: Building More Effective Processes by Integrating Process Modeling with Decision Modeling. CreateSpace Independent Publishing Platform, 2014.

[7] B. Von Halle and L. Goldberg, The Decision Model: A Business Logic Framework Linking Business and Technology. Auerbach Publications, 2009.

[8] M. Reichert and B. Weber, Enabling Flexibility in Process-Aware Information Systems. 2012.

[9] M. Dumas, W. Von der Aalst, and A. H. M. Hofstede, Bridging People and Software. Wiley-Interscience, 2005.

[10] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. Van Der Aalst, Process flexibility: A survey of contemporary approaches, vol. 10 LNBIP. 2008.

[11] H. a. Reijers, "Workflow flexibility: The forlorn promise," in Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, 2006, pp. 271–272.

[12] P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke, "A Comprehensive Approach to Flexibility in Workflow Management Systems," in Proceedings of the international joint conference on Work activities coordination and collaboration (WACC '99), 1999, pp. 79–88.

[13] R. Cognini, F. Corradini, S. Gnesi, A. Polini, and B. Re, "Research challenges in business process adaptability," in Proceedings of the 29th Annual ACM Symposium on Applied Computing - SAC '14, 2014, pp. 1049–1054.

[14] K. Swenson, N. Palmer, and B. Silver, Taming the Unpredictable. 2011.

[15] N. Palmer, "ACM Strategy and Business," in How Knowledge Workers Get Things Done, Future Strategies Inc., 2012, pp. 13–17.

[16] OMG, "Case Management Model and Notation," 2014. [Online]. Available: http://www.omg.org/spec/CMMN/1.0/.

[17] D. Sprovieri and S. Vogler, "Combining Business Processes and Cloud Services: A Marketplace for Processlets," in Proceedings of the 18th international Conference in Business Information Systems, 2015.

[18] F. Gottschalk, T. a C. Wagemakers, M. H. Jansen-Vullers, W. M. P. Van Der Aalst, and M. La Rosa, Configurable process models: Experiences from a municipality case study, vol. 5565 LNCS. 2009.

[19] A. Hallerbach, T. Bauer, and M. Reichert, "Capturing variability in business process models: the Provop approach," J. Software-Evolution Process, pp. 519–546, 2009.

[20] D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, and P. Plebani, "PAWS: A framework for executing adaptive web-service processes," IEEE Softw., vol. 24, no. 6, pp. 39–46, 2007.

[21] P. Völkner and B. Werners, "A decision support system for business process planning," Eur. J. Oper. Res., vol. 125, no. 3, pp. 633–647, 2000.

[22] R. Hull, E. Damaggio, R. De Masellis, F. Fournier, S. Maradugu, A. Nigam, P. N. Sukaviriya, and R. Vaculín, "Business Artifacts with Guard-Stage-Milestone Lifecycles : Managing Artifact Interactions with Conditions and Events," DEBS'11 - Proc. 5th ACM Int. Conf. Distrib. Event-Based Syst., pp. 51–62, 2011.

[23] C. Ayora, V. Torres, M. Reichert, B. Weber, and V. Pelechano, "Towards run-time flexibility for process families: Open issues and research challenges," in Lecture Notes in Business Information Processing, 2013, vol. 132 LNBIP, pp. 477–488.

[24] P. K. McKinley, S. M. Sadjadi, E. P. Kasten, and B. H. C. Cheng, "Composing adaptive software," IEEE Comput., vol. 37, no. 7, pp. 56–64, 2004.

[25] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," AI Commun., vol. 7, pp. 39–59, 1994.

[26] W. M. P. Van Der Aalst, M. Weske, and D. Grünbauer, "Case handling: A new paradigm for business process support," Data Knowl. Eng., vol. 53, no. 2, pp. 129–162, 2005.

[27] A. Murguzur, R. Capilla, S. Trujillo, Ó. Ortiz, and R. E. Lopez-Herrejon, "Context Variability Modeling for Runtime Configuration of Service-based Dynamic Software Product Lines," Proc. 18th Int. Softw. Prod. Line Conf. co-located Work., pp. 2–9, 2014.

[28] A. Murguzur, X. Carlos, S. Trujillo, and G. Sagardui, "Context-Aware Staged Configuration of Process Variants@Runtime," Adv. Inf. Syst. Eng. 26th Int. Conf. CAiSE 2014, pp. 241–255, 2014.

[29] A. Pierfranceschi, "Combining Process Modelling and Case Modeling," 2015.

[30] K. Hinkelmann and A. Pierfranceschi, "Combining Process Modelling and Case Modeling," in 8th International Conference on Methodologies, Technologies and Tools enabling e-Government MeTTeG14, 2014, pp. 1–11.

[31] OMG, "Decision Model and Notation," 2016. [Online]. Available: http://www.omg.org/spec/DMN/. [Accessed: 11-Apr-2016].